OBSERVACIONES DE LA PRÁCTICA

Juliana Rodríguez Morales 202421552

Maaria Clara Quijano 202420069

Estudiante 3 Cod XXXX

# **Ambientes de pruebas**

|  |  |
| --- | --- |
|  | Computador 1 |
| Procesador | 11th Gen Intel (R) Core(TM) i7-1165G7  @ 2.80 GHz |
| Memoria RAM (GB) | 32.0 GB |
| Sistema Operativo | Windows 11 Pro - 64 bits |

Tabla 1. Especificaciones del computador para ejecutar las pruebas de rendimiento.

Para realizar las siguientes pruebas, es importante que las ejecuten utilizando el archivo de datos **large**, ya que es con este conjunto donde realmente se puede observar el comportamiento y la eficiencia de los algoritmos en contextos más exigentes.

# **Ordenamientos Iterativos**

## **Resultados para ordenamientos iterativos con Array List**

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de la muestra | Insertion Sort  (Array List) | Selection Sort (Array List) | Shell Sort (Array List) |
| 0.50% |  |  |  |
| 5.00% |  |  |  |
| 10.00% |  |  |  |
| 20.00% |  |  |  |
| 30.00% |  |  |  |
| 50.00% |  |  |  |
| 80.00% |  |  |  |
| 100.00% |  |  |  |

Tabla 2. **Resultados en computador 1 para ordenamientos iterativos con Array List.**

## **Resultados para ordenamientos iterativos con Linked List**

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de la muestra | Insertion Sort  (Linked List) | Selection Sort  (Linked List) | Shell Sort (Linked List) |
| 0.50% |  |  |  |
| 5.00% |  |  |  |
| 10.00% |  |  |  |
| 20.00% |  |  |  |
| 30.00% |  |  |  |
| 50.00% |  |  |  |
| 80.00% |  |  |  |
| 100.00% |  |  |  |

Tabla 3. **Resultados en computador 1 para ordenamientos iterativos con Single Linked List.**

# **Ordenamientos Recursivos**

## **Resultados para ordenamientos recursivos con Array List**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Merge Sort (Array List) | Quick Sort (Array List) |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 4. **Resultados en computador 1 para ordenamientos recursivos con Array List.**

## **Resultados para ordenamientos recursivos con Linked List**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Merge Sort (Linked List) | Quick Sort (Linked List) |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 5. **Resultados en computador 1 para ordenamientos recursivos con Single Linked List.**

Una vez alla llenado l

# **Comparación de tiempo mejores algoritmos de ordenamiento**

**Complete esta sección únicamente cuando se le indique en las instrucciones. En este punto, deberá comparar el tiempo de ejecución entre el mejor algoritmo de ordenamiento recursivo y el mejor algoritmo de ordenamiento iterativo, los cuales debió haber identificado en el punto anterior.**

**Por favor, registre las mediciones en el espacio correspondiente de la tabla, especificando claramente:**

* **El nombre de cada algoritmo utilizado**
* **El tipo de estructura de datos empleada (ArrayList o SingleLinkedList)**

**Asegúrese de completar todos los campos solicitados en la tabla e incluir los nombres según corresponda, no es necesario que vuelva a correr las pruebas esta información la puede sacar de las tablas anteriores.**

## **Computador 1**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Algoritmo recursivo | Algoritmo iterativo |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 6. **Resultados en computador 1 mejores algoritmos**

# **Preguntas de análisis parte 1**

1. **¿Cómo varía el comportamiento de cada algoritmo con respecto al tamaño de los datos?**Para responder, agregue **líneas de tendencia** en las gráficas de las pestañas **04 a 08.** Observe si el crecimiento del tiempo de ejecución es **lineal, cuadrático o de otro tipo**, y relacione ese patrón con la **complejidad teórica esperada** de cada algoritmo.

**RTA: Por lo aprendido previamente, se puede concluir que insertion sort y selection sort tienen un tiempo de ejecución cuadrática ->O(n^2)**

**Por otro lado, merge sort y quick sort tienen complejidad O(n logn), que tiene un tiempo de ejecución lineal con curvatura.**

1. **¿Qué diferencias observas en el rendimiento de un mismo algoritmo al utilizar ArrayList frente a SingleLinkedList?**

Analice esta comparación en las pestañas **04 a 08**, donde se muestran ambos casos. Justifique su respuesta con base en las líneas de tendencia.

**RTA:** Array list tiende a tener mejor rendimiento debido a que es capaz de acceder a un dato en un índice dado, mientras que en linked list se debe recorrer toda la lista para llegar a una posición

1. **¿Qué algoritmo iterativo mostró el mejor comportamiento general en términos de tiempo y cantidad de datos en ambas estructuras de datos?**

Justifique su elección comparando las gráficas (04 **Insertion Sort)**, (05 **Selection Sort)** y (06 **Shell Sort** ) y apoyándose en las gráficas de las pestañas 2 y 3.

**RTA:** Shell sort normalmente tiene mejor comportamiento debido a que tiene un crecimiento menor que los otros.

1. **¿Qué algoritmo recursivo presentó el mejor desempeño general según los tiempos registrados y la forma de la línea de tendencia?**

Justifique su respuesta basándose en las pestañas 07 (Merge Sort) y 08 (Quick Sort) y apoyándose en las gráficas de las pestañas 2 y 3.

**RTA:** Merge sort es normalmente el mejor algoritmo recursivo,, tiene complejidad de O(nlogn) y no tiene un peor caso.

1. **Con base en su análisis visual y los datos recolectados, ¿cuáles algoritmos seleccionan como los “mejores” en cada categoría (iterativo y recursivo)?**

Indique también la estructura de datos en la que se desempeñó mejor cada uno.

**RTA:** Para el iterativo, el shell sort en array list es el mejor ya que se pueden acceder a las posiciones sin necesidad de recorrer todo n. Por otro lado, para los recursivos son

# **Preguntas de análisis parte 2**

1. **¿Cuál de los dos algoritmos seleccionados como “mejores” muestra un comportamiento más eficiente a medida que crece el tamaño de la muestra?**

Para responder, **agregue líneas de tendencia en la gráfica de la pestaña 09-Bests**, identifique cuál algoritmo crece más lentamente y relacione ese patrón con su complejidad teórica. Use esta comparación para justificar cuál es más escalable.

**RTA:**

1. **¿Qué estructura de datos resultó más favorable para cada algoritmo seleccionado como mejor?**

Analice si la ventaja se mantiene constante en todos los tamaños o solo en ciertos rangos.

**RTA:**

1. **¿Qué ventajas prácticas podría tener implementar el algoritmo ganador en aplicaciones reales donde se manejan grandes volúmenes de datos?**

**RTA: Si se implementa el algoritmo más eficiente, entonces en el caso de manejar grandes volúmenes de datos va a tener un menor tiempo de ejecución y requiere menos memoria.**

1. **Si se presentara una lista parcialmente ordenada, ¿cambiaría su elección de algoritmo? Explique por qué, con base en el comportamiento observado.**

**RTA:**